



## Ya diseñamos un círculo, ahora a darle vida

### Transcripción

[00:00] Hola, amigos y amigas. Vamos a continuar con nuestro curso de lógica de programación. En la última aula habíamos aprendido a diseñar círculos, circunferencias, donde hacíamos clic en la pantalla en nuestro programa y una circunferencia se dibujaba.

[00:20] Ahora, ¿cómo podemos hacer? Nuestro desafío en esta aula va a ser que nuestro programa automáticamente sea interactivo, que él cree esas animaciones en forma automática, que pueda dibujar circunferencias, cuadrados, cualquier figura geométrica que queramos, pero de forma automática, que el programa lo haga por sí solo.

[00:48] Para eso creé aquí un programa, un programa 4. Reaproveché una parte de nuestro código que estamos usando de padrón, nuestro Canvas, definiendo con el mismo tamaño aquí de largo y ancho, ancho y alto, nuestra variable pantalla, que recibe ese Canvas aquí en JavaScript.

[01:14] Estamos creando una variable pincel, que es como si fuera nuestro dibujador, la cual recibe el contexto del Canvas en dos dimensiones, que es donde vamos a diseñar. Estamos definiendo nuestro Canvas en color grey plomo. Le vamos a poner un light grey para que sea un plomo claro y no un plomo oscuro. Y bueno, nuestro rectángulo padrón de 600 x 400.

[01:48] Primeramente, vamos a abrir aquí nuestro programa, en este caso, es el cuarto, y aquí está nuestro Canvas, nuestra pantallita. Creo que no guardé. Ahora sí, está claro. Perfecto. Vamos a crear una función, como ya lo hemos

dicho antes, para dibujar círculos, circunferencias. Voy a llamarlo `disenharCircunferencia`.

[02:22] Aquí, yo por comodidad siempre acostumbro colocar la H cuando tenemos la Ñ, para no colocar un carácter específico del lenguaje español, coloco H o N. Hay otras personas que simplemente ignoran y colocarían `disenar`. A mí particularmente me gusta usar esa H ahí intermedia como si fuera portugués.

[02:49] `DisenharCircunferencia`. Entonces, va a recibir algún parámetro esa nuestra función. ¿Cuáles serían esos parámetros? Básicamente la coordenada, la coordenada horizontal y vertical que queremos construir aquí, y el tamaño de esta circunferencia, el radio de la circunferencia. En este caso el radio es el radio dividido entre 2 o la mitad del diámetro.

[03:24] Y nuestra función de circunferencia, ¿qué instrucciones necesita tener? Vamos a reutilizar nuestro pincel, que ya lo habíamos definido arriba y vamos a definir el color. Lo primero que hacemos es definir el color que va a tener nuestra circunferencia, vamos a continuar con el azul.

[03:47] Luego iniciamos un camino, `beginPath`, que es como decirle: "Voy a comenzar a graficar y te voy a pasar luego las coordenadas". Luego teníamos, ya graficábamos nuestra circunferencia. Teníamos la función `arc`, para graficar, la cual recibía cinco parámetros. En este caso, ¿qué va a recibir, si recordamos de las anteriores aulas o las anteriores secciones?

[04:29] Recibe las coordenadas, son los dos primeros parámetros, coordenada horizontal y vertical, dónde queremos que empiece, cuál va a ser el punto inicial de la circunferencia, cuál va a ser el centro de la circunferencia, después pasábamos el radio, el tamaño de nuestra circunferencia y por último el ángulo inicial y el ángulo final en radiales.

[04:51] Entonces, como aquí hemos definido los parámetros que va a recibir, que son X e Y, aquí vamos a pasar ese parámetro, y el radio de nuestra

circunferencia también va a ser un parámetro que le vamos a pasar a esa función al momento de llamarla. Bueno, el parámetro del ángulo inicial es 0 y dos veces pi, recordando que pi es 3.14.

[05:19] Sin embargo, ya JavaScript tiene configurado dentro de las operaciones matemáticas el + que ya lo hemos utilizado para redondear números, redondear valores, también tenemos el Math.PI, recordando que 3.14 es el número de pi, es la relación entre un diámetro de una circunferencia, y el perímetro de la misma.

[05:48] Es un número constante que siempre es el mismo, 3.1416 y es un decimal periódico. Y para no estar escribiendo 3.14 y hasta para ser más precisos con ese número, existe una función ya padrón en JavaScript que en este caso es Math.PI que la podemos utilizar.

[06:10] Y por último, tenemos nuestra función que le dice: "llenalo, llename esa circunferencia". Fill. Llename la circunferencia. Perfecto. Y aquí llamamos a esa función `disenhar.Circunferencia` y vamos a pasar los tres parámetros, que empiece en la posición 20, 20 y vamos a pasar aquí el tamaño del radio 10. Voy a guardar, actualizamos. Algún error hay por aquí.

[06:56] Voy a ir a mi consolidador para ver si me ayuda, y me dice en la línea 11. En la línea 11 tenemos algún error. Aquí. Me olvidé de algo muy importante que decirles que esa es una función. Tenemos otro error en la línea 14. `Circunferencia`, pincel, aquí. En este caso, la consola de programación, tenía dos N, me ayuda a encontrar e identificar esos errores. Y aquí está ya mi circunferencia graficada.

[07:59] Esos errores no son a propósito, justamente es porque al momento que uno va haciendo rápidamente el programa, nos van apareciendo los errores sin que nos demos cuenta, pero nos acostumbramos a usar la consola. Todos los compiladores tienen una consola donde uno puede verificar los errores y eso, en este caso, el curso va a ayudar a que ustedes vean cómo se hace ese

tratamiento y esa identificación de errores que a veces puede demorar horas para identificar un error en su código.

[08:30] Pero siguiendo las buenas prácticas y teniendo sobre todo mucho cuidado, podemos evitar esos errores. Entonces, aquí tenemos nuestra bolita, nuestra circunferencia graficada, podemos mandar una con un tamaño mayor y mudando, cambiando un poco las coordenadas. Tenemos otra aquí.

[09:00] Pero bueno, lo que en realidad queremos es que esta circunferencia se mueva, como si fuera una víbora, como si fuera el juego de la viborita, se mueva en la parte superior de la pantalla hasta terminar hasta acá. Eso es lo que vamos a ver cómo hacer en nuestro próximo video.